**Google OAuth**

* Create a app in Django and configure it in the settings.py file
* Go to google cloud console
* Create new project “project name : Django-google-oauth”
* Select the project
  + Api and services
  + Oauth consent screen
    - Oauth consent screen 🡺 external
    - Fill app name “Django Tutorial”
    - User support email [yokeshanand762001@gmail.com](mailto:yokeshanand762001@gmail.com)
  + Scopes
    - Add or remove scopes 🡺 adding user info and user profile
  + Save and go to dashboard
  + Go to credentials
    - Create credentials
      * Oauth client id 🡪
        + Application type : Web application
        + Name : Django (The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.)
        + Authorized redirect urls:

<http://127.0.0.1:8000>

<http://127.0.0.1:8000/accounts/google/login/callback>

* + - * received client id: 684025040233-l6elp5tarpck4bnu0moik18va80h79ij.apps.googleusercontent.com
      * received client secret: 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
      * download and save the json on the project folder
* After the above process go to vs code
  + In the settings.py
    - above the installed apps add site\_id = 1
    - 'django.contrib.sites',  #  allows you to associate your Django project with multiple sites
    - 'allauth', #'allauth' and 'allauth.account':  are used to implement user authentication and registration systems with features like email confirmation, password reset, and more. They are highly customizable and provide a solid foundation for user management.
    - 'allauth.account',
    - 'allauth.socialaccount', # 'allauth.socialaccount' extends 'allauth' to enable social authentication, making it easy for users to sign in with their social media accounts.
    - 'allauth.socialaccount.providers.google', #is a specific provider package that adds Google authentication to your site. By integrating it, you allow users to sign in or register using their Google credentials.
  + Specify variables to our social account provider on settings.py
    - import os
    - SOCIALACCOUNT\_PROVIDERS = {
    - "SCOPE":[
    - "profile",
    - "email"
    - ],
    - "AUTH\_PARAMS": {
    - "access\_type": "online"
    - },
    - }
    - }
  + At the end of the settings.py add the authentication backend
    - AUTHENTICATION\_BACKENDS = ( # here we are specifying that we are using both  the django backend and allauth backends
    - "django.contrib.auth.backends.ModelBackend",
    - "allauth.accounts.auth\_backends.AuthenticationBackend"
    - )
  + Add login and logout redirect urls
    - LOGIN\_REDIRECT\_URL = "/"
    - LOGOUT\_REDIRECT\_URL = "/"
* On the urls.py in the overall project folder
  + Include path for accounts and user app
    - path('accounts/', include("allauth.urls")) # this is for login, logout, password reset etc..
    - path('/', include('users.urls')) # for the user app
* On the urls.py in the app folder
  + Create urls.py file and add these urls
    - urlpatterns = [
    - path("", views.home),
    - path("logout", views.logout\_view)
    - ]
* On the views.py
  + Create those views mentioned in urls.py
    - from django.shortcuts import render, redirect
    - from django.contrib.auth import logout
    - # Create your views here.
    - def home(request):
    - return render(request, "home.html")
    - def logout\_view(request):
    - logout(request)
    - return redirect('/')
* To render home.html
  + On the app folder create folder named templates
  + In templates create home.html file
  + On home.html
    - <!doctype html>
    - <html lang="en">
    - <head>
    - <title>Google Sign In</title>
    - <!-- Required meta tags -->
    - <meta charset="utf-8">
    - <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    - <!-- Bootstrap CSS -->
    - <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    - </head>
    - <body>
    - {% load socialaccount %}
    - {% comment %} this will enable to use the login urls {% endcomment %}
    - <h2>Google Login</h2>
    - <a href="{% provider\_login\_url 'google' %}?next=/">Login With Google</a>
    - {% comment %} this will provide the url for google signin  {% endcomment %}
    - <!-- Optional JavaScript -->
    - <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    - <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
    - <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
    - <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
    - </body>
    - </html>
* Then on the terminal run makemigrations and migrate to replicate changes to the database
* Go to admin and delete the existing sites and create new site with http://127.0.0.1:8000
* Go to social application and create social application with client id and secret key
* This will be the final html
  + <!DOCTYPE html>
  + <html lang="en">
  + <head>
  + <meta charset="UTF-8">
  + <meta name="viewport" content="width=device-width, initial-scale=1.0">
  + <title>Google Sign In</title>
  + </head>
  + <body>
  + {% load socialaccount %}
  + {% comment %} this will enable to use the login urls {% endcomment %}
  + <h2>Google Login</h2>
  + <a href="{% provider\_login\_url 'google' %}?next=/">Login With Google</a>
  + {% if user.is\_authenticated %}
  + <p>You are signed in as {{user.email}}</p>
  + <a href="logout">Logout</a>
  + {% endif %}
  + {% comment %} this will provide the url for google signin  {% endcomment %}
  + </body>
  + </html>